(2)

AD-A227 912

August 23, 1989 Final

Multiprotocol Gateway SDL Description C-DCA100-87-C-0063

Booz-Allen & Hamilton, Inc.
4330 East-West highway
Bethesda, MD 20814

National Communications System
Office of Technology & Standards
Washington, DC 20305-2010 NCS TIB 89-7

Point of Contact: Dennis Bodson, 202-692-2124

Approved for public release; distribution unlimited

Present telecommunication networks use one of many different protocol suites to interconnect their services. Networks employing different protocols cannot communicate with one another because the diverse protocols are incompatible. To address this protocol incompatibility issue it was recommended to develop mechanisms, such as gateways that provide protocol conversion, to interconnect networks using diverse protocols. These devices will act as translators between the two networks while maintaining communications with each network in its native language.

Gateways
ISDN
Protocols 39

Unclassified Unclassified Unclassified

NCS TIB 89-7

NATIONAL COMMUNICATIONS SYSTEM

TECHNICAL INFORMATION BULLETIN

89-7

MULTIPROTOCOL GATEWAY SDL DESCRIPTION

AUGUST 23, 1989

OFFICE OF THE MANAGER
NATIONAL COMMUNICATIONS SYSTEM
WASHINGTON, D.C. 20305

MULTIPROTOCOL GATEWAY
SDL DESCRIPTION

AUGUST 23, 1989

BOOZ•ALLEN & HAMILTON INC.
4330 EAST WEST HIGHWAY
BETHESDA, MD 20814

PREPARED FOR
OFFICE OF THE MANAGER
NATIONAL COMMUNICATIONS SYSTEM
8TH & SOUTH COURTHOUSE ROAD
ARLINGTON, VA 22204

TABLE OF CONTENTS

LIST OF EXHIBITS

1.0 INTRODUCTION

The Office of the Manager, National Communications System (OMNCS), has been tasked to ensure the provision of National Security and Emergency Preparedness (NS/EP) telecommunications for the Federal Government under all conditions. In support of this task, the OMNCS is working to promote standards within the data communication and telecommunication fields to increase the interoperability of diverse communication systems. These interoperability efforts improve the NS/EP communication capabilities of the nation by providing additional interconnectivity among Federal telecommunication assets. One such effort is determining the feasibility of using the excess transmission capacity of the Integrated Services Digital Network (ISDN) out-of-band signaling channels to reconstitute disrupted networks.

1.1 BACKGROUND

Present telecommunication networks use one of many different protocol suites to interconnect their services. Networks employing different protocols cannot communicate with one another because the diverse protocols are incompatible. To address the protocol incompatibility issue, the OMNCS recommended developing mechanisms, such as gateways that provide protocol conversion, to interconnect networks using diverse protocols. These devices will act as translators between the two networks while maintaining communications with each network in its native language.

In the data communications field there are a number of prevalent standards for supporting end-to-end communications. Some of the existing sets of protocols that the OMNCS has identified to provide interoperability are:

- The International Telegraph and Telephone Consultative Committee's (CCITT) X.25 Recommendations
- The CCITT's recommendations for Integrated Services Digital Networks (ISDNs)
- The U.S. Department of Defense (DOD) developed Transport Control Protocol (TCP)/Internet Protocol (IP).

Since these standards are expected to coexist during the near future, gateways are needed that will allow users of these diverse protocols to communicate.

Earlier OMNCS efforts focused on specifying a gateway between X.25 and TCP/IP. Both a gateway development and description [1] and a specification and description language (SDL) document [2] were developed. These documents provide information to begin software development of an X.25-TCP/IP

Gateway that would allow users of the two protocols to exchange information.

More recent OMNCS efforts have focused on developing a network component to allow X.25 end user traffic to use future ISDNs as transport networks to interconnect disrupted portions of their packet-switched networks. This work investigated the use of the excess transmission capacity of ISDN out-of-band signaling channels to reconstitute disrupted networks. To assess this capability, a comparison of the two protocols was prepared. This was followed by a definition of protocol operating states and transitions among states needed to control and implement an X.25-ISDN Gateway. Based on these preliminary operating states, a functional SDL document for a X.25-ISDN Gateway was developed. The X.25-ISDN Gateway has been implemented and its capabilities demonstrated. The X.25-ISDN Gateway and demonstration are described in X.25-ISDN Gateway High Level Language Program Demonstration, National Communications System, May 12, 1989.

1.2 PURPOSE

This document contains an SDL description for a gateway designed to interconnect CCITT X.25 end users, Defense Data Network (DDN) X.25 (TCP/IP) end users, as well as CCITT X.25 and DDN X.25 (TCP/IP) end users, over the ISDN D channel. The gateway description provides the foundation from which the gateway implementation is to be directed. The specification outlines the internal functionality of gateway components, their modules, and the interrelationship between these modules.

1.3 PROJECT REFERENCES

The following documents have been used in the development of this document:

1. X.25/PLP/TCP Gateway Development and Description. National Communications System. February 1986.

2. X.25/PLP-DDN/TCP Gateway SDL Technical Specification. National Communications System. June 1986.

3. X.25 and ISDN Packet Mode State Transition Tables. National Communications System. November 1987.

4. X.25-ISDN Gateway State Transition Tables. National Communications System. May 1988.

5. Data Communications Networks: Interfaces. Recommendations X.20-X.32. CCITT, Vol. VIII - Fascicle VIII.3. 1984.

6. Integrated Services Digital Network (ISDN). Series I Recommendations. CCITT, Vol. III - Fascicle III.5. 1984.

7. X.25-ISDN Gateway SDL Description. National Communications System. August 1988.

8. Functional Specification and Description Language (SDL) Recommendations Z.101 - 104. CCITT, Vol. VI - Fascicle VI.10. 1984.

9. Defense Data Network X.25 Host Interface Specification. Defense Communications Agency. December 1983.

10. Internet Protocol. MIL-STD-1777, Defense Communications Agency, J110. 12 August 1983.

11. Transmission Control Protocol. MIL-STD-1778, Defense Communications Agency. 12 August 1983.

12. X.25-ISDN Gateway High Level Language Program Demonstration. National Communications System. May 12, 1989.

The foundation of the multiprotocol gateway is composed of the CCITT protocol specifications (5 and 6) and the DDN protocol specifications (9 though 11). Other references include documents relating to earlier efforts in support of a X.25-TCP/IP gateway (1 and 2) and efforts involving the X.25-ISDN gateway (3 and 4). The methodology and syntax used to specify the gateway is based on a separate series of CCITT recommendations -- Z series (8). An overview of the SDL is provided below.

1.4 SDL OVERVIEW

The SDL syntax has been developed by CCITT for unambiguously describing the behavior of telecommunication systems. A system is defined in SDL through a set of interconnected blocks. The blocks are connected through unidirectional channels. Each system block describes the functional operation of a portion of the system. Further refinement can be obtained by dividing a block into sub-blocks, which communicate via sub-channels. This successive decomposition is useful for expressing system hierarchy.

In the SDL syntax, each system block contains one or more processes. A process is defined as a communicating finite-state machine composed of multiple processing states. A process is usually used to represent a function of a block, such as setting up or tearing down a connection. Processes communicate through signals, and reception of any one of a predefined set of input signals triggers state transitions within the process.

Processes can also be decomposed into sub-processes to show greater detail. Further refinement can be obtained by partitioning the sub-processes into procedures. The final degree of detail is obtained by splitting the procedures into macros.

In this document, the multiprotocol gateway SDL specification is presented in terms of process diagrams. The process diagrams provide the level of fidelity necessary to support gateway software development. This is consistent with the I series of recommendations, which also defines the protocols at the process level. The basic symbols used in the process diagrams, including a brief definition of each symbol, are shown in Exhibit 1-1. Descriptive names are given to all SDL symbols that specify the function of the symbol. Additional SDL information can be obtained from reference 8.

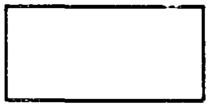
1.5 ORGANIZATION

This report is composed of four sections. Section 2 focuses on the top level architecture with a discussion and explanation of each of the defined gateway modules. Also included in this section are the environment that the gateway will be operating in and a discussion of the likely system users and user applications. The elements that compose the gateway are discussed in Section 3. This includes a definition of required hardware devices, software, interfaces, and security measures. The SDL specifications for the multiprotocol gateway are presented in Section 4.

EXHIBIT 1-1

SDL Symbols And Definitions

Signal route symbol-- A signal route symbol leads either from one process to another, or from a process to the origin end of a channel, or from the destination end of a channel to a process.

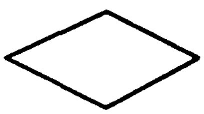


Task symbol-- A task is used to represent operations on data performed on a transition, with the exception of output signal generation and decisions.

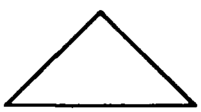
Input symbol-- An input symbol attached to a state means that if the signal named within the input symbol arrives while the process is in this state, the transition that follows the input symbol should be interpreted.

Output symbol-- An output symbol represents the sending of a signal from one process to another.

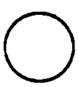
State symbol-- A state is a point in the process where no actions are being performed but where the input queue is monitoring for the arrival of incoming signals.



Decision symbol-- A decision is an action within a transition that asks a question regarding the value of data items available to the process at the instant of executing the action.



Option symbol-- The option symbol is used to describe several alternative process behaviors with one diagram.



Connector symbol-- Any flow line may be broken by a pair of associated connectors, with the flow assumed to be from the out-connector to the in-connector.

Return symbol-- This symbol implies a return to the initial process state.

Stop symbol-- This symbol implies the end of the process.

2.0 GATEWAY TOP LEVEL ARCHITECTURE

The multiprotocol gateway described in this report will be designed to support the interconnection of CCITT X.25 end users, DDN X.25 (TCP/IP) end users, as well as CCITT X.25 and DDN X.25 (TCP/IP) end users, over the ISDN D channel. Consistent with the mission of the OMNCS, the gateway is designed to support emergency communications. This requires that the gateway be transportable and use hardware that is readily available. For these reasons, the gateway will be developed for a TBD based microcomputer using an INTEL 386 microprocessor. However, the modular design approach to the gateway is applicable to the design of larger gateways, such as a Digital Equipment Corporation MicroVAX II based gateway that could support multiple connections at a faster throughput rate.

2.1 ENVIRONMENT

The environment in which the multiprotocol gateway will be operating is shown in Exhibit 2-1. As pictured, the environment is divided into four sections: the X.25 network, the DDN (TCP/IP) network, the ISDN, and the gateway. For the gateway to be useful it is necessary that the network users operate in their native mode. It will be the gateway's responsibility to make up for all incompatibilities. Network users will not be required to have any additional equipment when communicating across networks.

The gateway can be subdivided into the following four components:

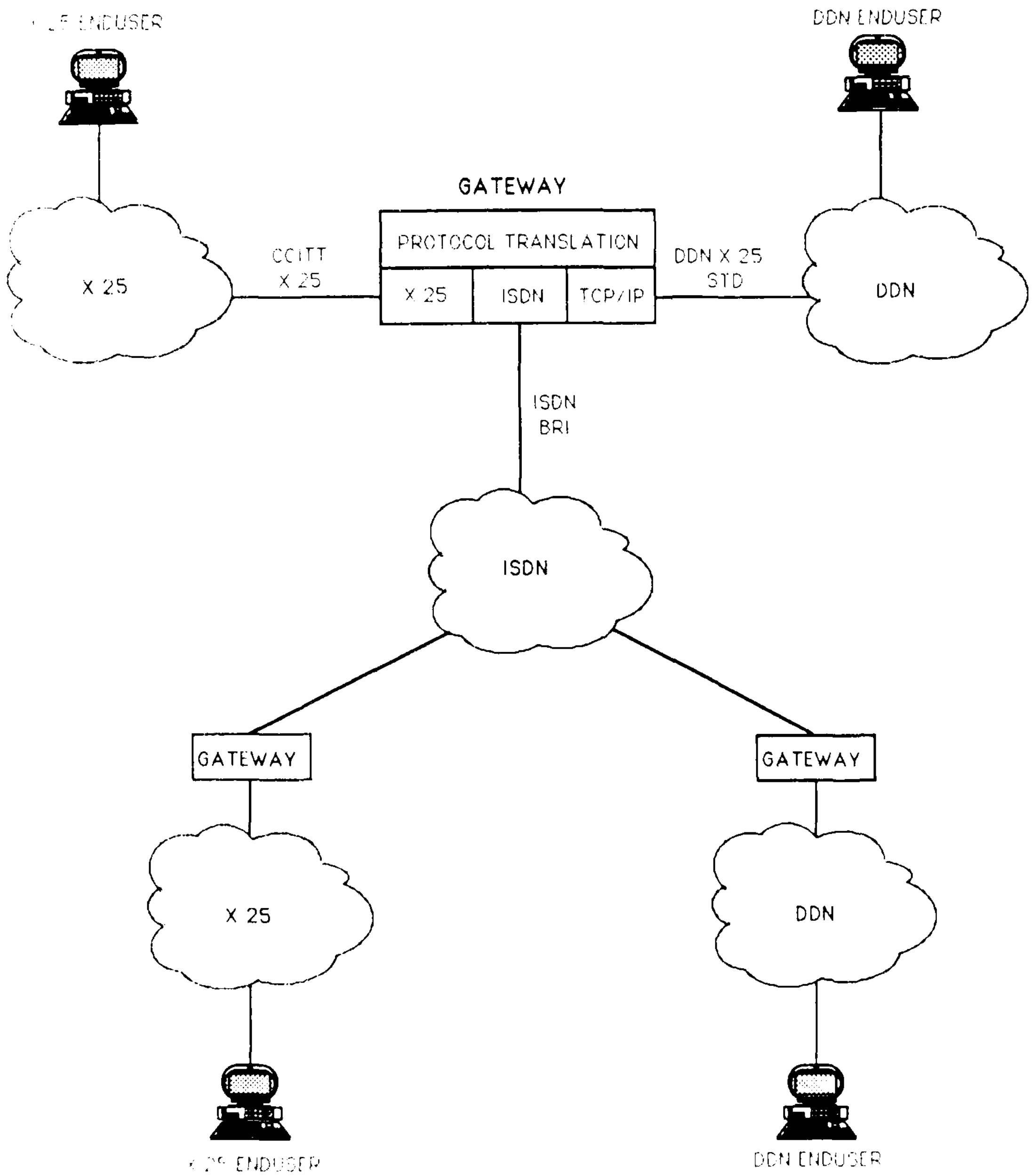
- CCITT X.25 network interface
- DDN X.25 (TCP/IP) network interface
- ISDN Terminal Adaptor (TA) interface
- Protocol Translation.

The CCITT X.25 network interface can be configured to appear as a standard data terminal equipment (DTE) device connected to an X.25 network or it can be configured to appear as a standard data circuit-terminating equipment (DCE) device supporting an X.25 end user. The CCITT X.25 network interface will perform all addressing, flow control, and call setup/clearing that are normally part of the X.25 protocol. The CCITT X.25 hardware and software will be purchased as off-the-shelf, commercially-developed products.

Similarly, the DDN X.25 (TCP/IP) network interface can be configured to appear as a standard DTE device connected to the DDN or it can be configured to appear as a standard DCE device supporting a DDN X.25 end user. The DDN X.25 network interface will perform all addressing, flow control, and call setup/clearing that are normally part of the TCP/IP protocol. The DDN X.25 (TCP/IP) hardware and software will be purchased as off-the-shelf, commercially-developed products.

EXHIBIT 2-1

External Gateway Environment



The ISDN TA interface provides the basic rate interface (BRI) to the ISDN. The ISDN BRI supports two 64 kilobits per second (kbps) circuit-switched channels (referred to as B channels) and one 16 kbps packet data channel (D channel). The D channel serves two main purposes. First, it carries common-channel signaling information to control circuit-switched calls on associated B channels at the user interface. In addition, the D channel may be used for packet-switching or low-speed (e.g., 100 bps) telemetry at times when no signaling information is waiting. The multiprotocol gateway uses the excess bandwidth in the D channel to provide packet-switched transport services. The ISDN TA hardware and software will be purchased as off-the-shelf, commercially-developed products.

The protocol translator will be the interface between the three end user types. It will perform packet translations, maintain address tables, track status of the connections, and provide the man-machine interface with the gateway. The protocol translator will be developed and implemented using the Microsoft "C" programming language.

2.2 SUBSYSTEM SPECIFICATION

The gateway is structured based on the specific functional processes performed internal to the gateway. A block diagram of the gateway functional processes is shown in Exhibit 2-2. The gateway is composed of ten functional processes: gateway driver, protocol startup, link answer setup, call establishment, data transfer, statistics display, link shutdown, ISDN module, X.25 module, and the TCP/IP module. The ISDN, X.25, and TCP/IP modules perform all of the network-specific functions associated with the gateway. The other seven modules perform gateway-specific functions. These functional processes are described in more detail below.

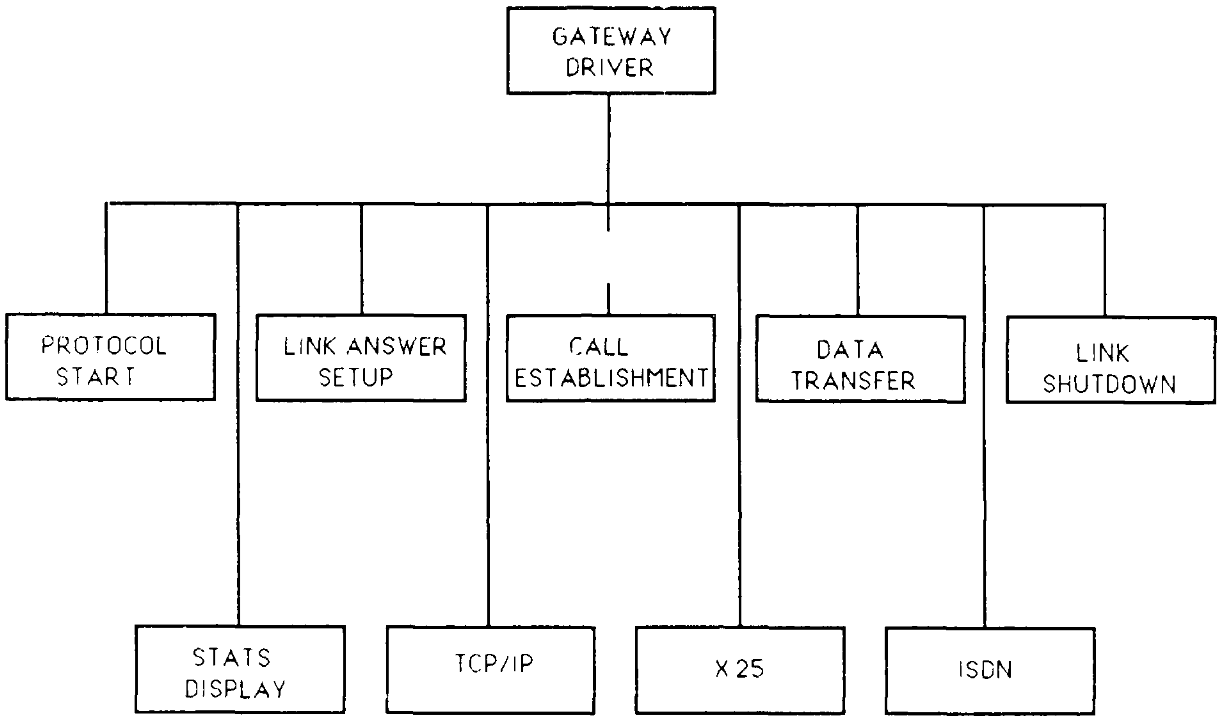
2.2.1 Gateway Driver

Control of all gateway activity is provided through this process. The gateway device driver communicates to each of the other processes and controls which processes are activated. Based on information primitives received from the supporting network interface cards, the process controls the transitions between operating states. The gateway device driver also tracks the status of the call, gathers call statistics, and monitors the various network interfaces.

2.2.2 Protocol Startup

This process is responsible for initiating gateway operation. This includes loading the gateway software and down-loading the X.25, TCP/IP, and ISDN software. Once the gateway is capable of starting a connection, the gateway driver is notified.

EXHIBIT 2-2
GATEWAY BLOCK DIAGRAM



2.2.3 Link Answer Setup

This process is responsible for resetting the gateway variables and placing the X.25, TCP/IP, and ISDN network modules in a listen state. In the listen state, the network modules wait for a connection attempt from their respective networks and then notify the gateway driver of any attempts.

2.2.4 Call Establishment

Upon reception of a connection request packet, the Call Establishment process establishes a connection with the specified user on the destination network and returns a connection status message to the gateway driver. This process includes address translation and link establishment.

2.2.5 Data Transfer

After connection establishment, the Data Transfer process accepts data packets from the respective network module and forwards the data to the destination network output buffer for transmission to the destination end user. The Data Transfer process also includes the repacketization and connection termination functions.

2.2.6 Statistics Display

The Statistics Display process provides run-time statistics and statistics on the performance of the gateway to the gateway operator.

2.2.7 Link Shutdown

The Link Shutdown process performs the shutdown procedures upon command from the gateway operator. The link shutdown process controls a sequenced shutdown of all gateway and connection processes.

2.2.8 ISDN Protocol Suite

This process performs all physical, data link, and network layer operations specified for an ISDN BRI. The physical layer covers the physical link between devices and the rules by which bits are passed from one to another. The data link layer attempts to make the physical link reliable and provides the means to activate, maintain, and deactivate the link. The network layer provides for the transparent transfer of data between transport entities. The process receives information packets from the ISDN and after completing the required processing, forwards the appropriate information to the gateway device driver for further processing by the gateway. In addition, information primitives sent from the gateway are directed to the ISDN via the ISDN protocol suite process.

2.2.9 X.25 Protocol Suite

This process is analogous to the ISDN protocol suite process with respect to the X.25 protocol.

2.2.10 TCP/IP Protocol Suite

This process performs all physical, data link, network layer, and transport layer operations specified for a DDN (TCP/IP) X.25 user. The transport layer (TCP) ensures that data units are delivered error free, in sequence, with no losses or duplications. This process is analogous to the X.25 protocol suite with respect to DDN X.25 and the TCP/IP protocol.

2.3 SYSTEM USERS

Two types of users are defined in this multiprotocol gateway specification. One type of user is a real-time X.25 or DDN end user making use of the additional transport facilities offered by the gateway. The second type of user is the passive run-time operator for the gateway.

The real-time subscriber uses the gateway's capabilities when access across the user's principal transport mechanism has been disrupted. The gateway provides connectivity through the excess capacity of existing ISDNs. The real-time subscriber can obtain access to the gateway's capabilities through one of two modes of connection. The first mode is through a direct, hard-wire DTE-to-DCE connection to the gateway. In the second mode, the gateway is a defined DTE endpoint on a network (X.25 or DDN) and an end user can access the gateway by routing a call to the gateway through the operational portion of the associated network.

The gateway described in this specification does not support protocol translation for session level and above. The applications used by each of the end users (X.25 to TCP/IP) during a call must be compatible. The gateway does not modify the protocol data units above the transport level in any way. As a next step in the multiprotocol development, an application protocol translation capability (e.g., FTP-FTAM, SMTP-X.400, TELNET-Virtual Terminal Protocol) could be incorporated into the gateway.

The second type of system user is the gateway operator. The operator controls all gateway functionality including startup, shutdown, and status monitoring. In addition, the gateway operator controls the loading and maintenance of the addressing and routing table information.

2.4 ACCURACY AND VALIDITY

The multiprotocol gateway will be designed to ensure the integrity and validity of all information presented for processing. Information passed through the gateway will be maintained with 100 percent accuracy and validity. Information transferred through the gateway will not be manipulated other than to strip and attach the necessary protocol header information. Information contained in the data fields of incoming packets will be copied from one packet type to another. In addition, the inherent error detection and correction algorithms used in the CCITT X.25, X.31, and TCP/IP protocols will work to reduce the probability of errors during transmission across the physical links.

2.5 TIMING

Gateway timing is based on the defined hardware and software configuration that supports at a minimum the following physical links:

- One 4.8 kbps X.25 connection
- One ISDN BRI operating at 16 kbps on the D channel
- One DDN X.25 (TCP/IP) connection operating at 4.8 kbps.

The multiprotocol gateway will meet the following timing goals:

- Total processing delay (defined to be the sum total of the Master Processing Unit [MPU] + two times the average Network Interface Card delay) less than or equal to .072 seconds
- Total queueing delay less than or equal to .108 seconds
- Total gateway delay less than or equal to .20 seconds.

2.6 FLEXIBILITY

The multiprotocol gateway will be designed in a structured, top-down manner to ensure easy expansion of operating capabilities. The gateway design will provide the ability to support additional physical connections to the gateway as well as the required software and specialized Network Interface Cards (NIC) needed to support additional protocol translations.

3.0 SYSTEM CONFIGURATION

This section describes the overall multiprotocol gateway configuration. The equipment to be used in the implementation of the gateway and the supporting software is described. The gateway external interfaces are defined and a discussion of present and future security requirements is presented.

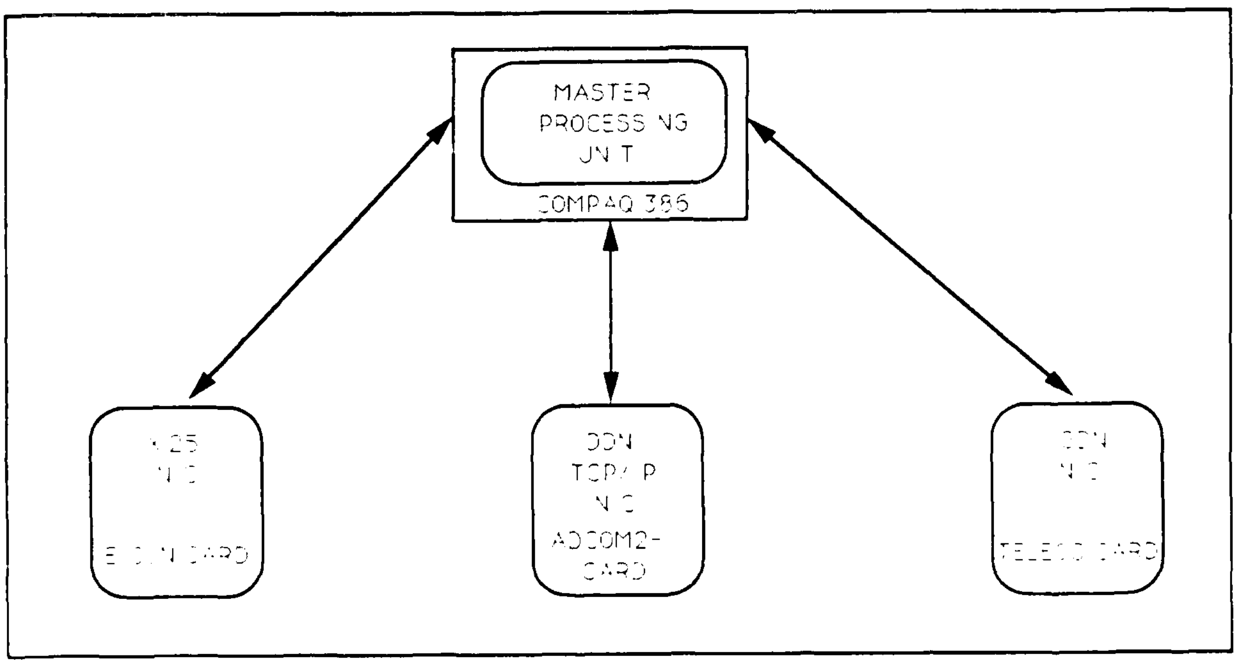
3.1 EQUIPMENT CONFIGURATION

The multiprotocol gateway equipment environment is composed of two types of components; a Master Processing Unit (MPU) and a Network Interface Card (NIC). Each component is designed around an independent micro-processor unit, supporting memory, and input/output controllers.

Each operational gateway configuration is composed of a single MPU controlling gateway operation and three NICs to provide the interface for each of the physical connections supported. Exhibit 3-1 shows the four basic gateway components and their interrelationships.

EXHIBIT 3-1

Multiprotocol Gateway Component Configuration



3.1.1 Master Processing Unit

The MPU controls all gateway operations, including:

- User interface
- Protocol and packet translations
- Address mappings
- Packet routing
- Statistics gathering and error reporting
- Packet buffering
- Real-time scheduling of multiple independent processes
- Communications across components.

The MPU selected for the multiprotocol gateway is based on a Compaq 386 supermicrocomputer with the following characteristics:

- 32 bit microprocessor; based on an INTEL 386 chip
- 4 megabyte (Mb) main memory
- 20 megahertz (MHz) internal clock
- Adjustable cache memory
- 120 Mb direct access storage
- 1.2 Mb floppy disk.

This configuration will support system development and test as well as the real-time operation of the gateway.

3.1.2 Network Interface Cards

Three types of NICs are to be used in the multiprotocol gateway. The first NIC provides the interface to the CCITT X.25 network, the second supports the interface to the DDN X.25 (TCP/IP) network, and the third provides the terminal adaptor capabilities required for the ISDN. The NICs perform the unique protocol processing functions required of the physical, link, and network layers of the specific networks being connected. The DDN X.25 NIC also supports the transport layer functions provided by TCP. All of the NICs are personal computer adaptor boards and interface directly to the MPU's input/output bus.

The EICON X.25 PC card has been selected for the CCITT X.25 NIC. The card has the following characteristics:

- 32 bit microprocessor; based on Motorola 68008 chip
- 512 Kilobytes (KB) ranaom access memory (RAM)
- 3.68 Mhz internal clock
- 1 Z8530 serial communications controller
- 4 direct memory access channels.

All processing is controlled through software downloaded from the MPU, requiring no off-line program storage.

The DDN X.25 (TCP/IP) NIC selected is the Frontier Technologies Corporation's AdCom2-I Intelligent Communication Controller. The card has the following characteristics:

- 80188 microprocessor
- 512 KB RAM
- Z80 based communications controller
- 4 direct memory access channels

All processing is controlled through software downloaded from the MPU, requiring no off-line program storage.

The ISDN NIC selected is the TELEOS B100PC Communications Coprocessor. The B100PC Communications Coprocessor is an IBM PC/XT/AT-compatible basic rate (2B+D) adapter, providing ISDN S/T interface compatibility in accordance with CCITT recommendations at the physical, data link, and network layers. The card has the following characteristics:

- 68HC000 processor
- 512 KB RAM
- 12.288 MHz clock
- 4 direct memory access channels.

Similar to the X.25 NIC, all processing is controlled with real-time software that is downloaded from the MPU and requires no off-line storage. The TELEOS equipment has been used previously in the development and implementation of the X.25-ISDN Gateway (12).

3.2 SUPPORT SOFTWARE CONFIGURATION

Two types of software support tools are to be used in the development and implementation of the multiprotocol gateway. One type of tool provides for the real-time software development while the second supports software testing and debugging. These tools are used to support the development of protocol processing and overall gateway control software used in the MPU. The two tools are intended exclusively for use on the MPU. No software modifications are deemed necessary to the NICs used in the gateway. The software development tools include the MS-DOS operating system, a line editor, and MicroSoft's 'C' programming language compiler.

3.3 INTERFACES

The multiprotocol gateway provides four external interfaces: a gateway/CCITT X.25 interface, a gateway/DDN X.25 interface, a gateway/ISDN interface, and a Man-Machine Interface (MMI). These interfaces are shown in Exhibit 3-2.

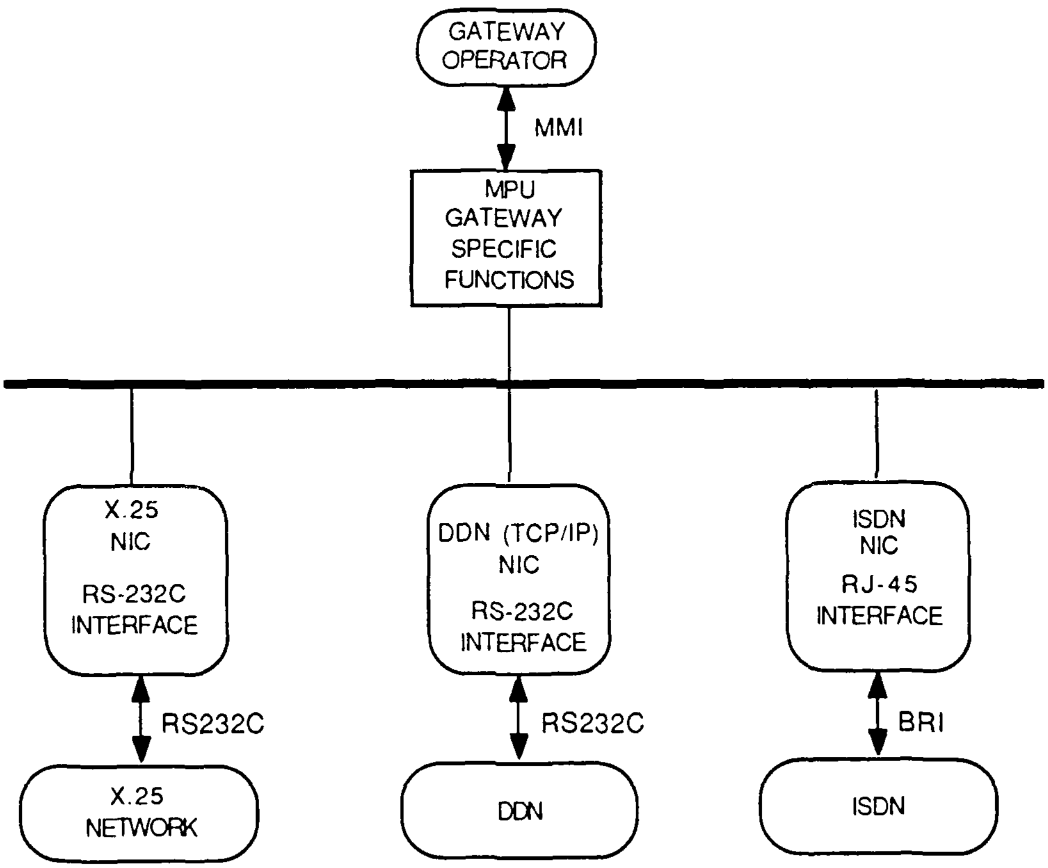
3.3.1 Gateway/CCITT X.25 Network Interface

The gateway/CCITT X.25 interface is provided through the EICON NIC. The EICON NIC supports the following:

- RS-232-C network interface
- 32 virtual circuits
- Nine concurrent host sessions

EXHIBIT 3-2

Multiprotocol Gateway External Interfaces



- Selectable, internal/external clocking
- Data Rates Supported: 2400, 4800, 9600, 19200, 38400, 48000, 56000, 57600, 64000.

3.3.2 Gateway/DDN X.25 Interface

The gateway/DDN X.25 interface is provided through the AdCom2-I NIC. The AdCom2-I NIC supports the following:

- RS-232-C network interface
- 32 virtual circuits
- Selectable, internal/external clocking
- Data Rates: 2400, 4800, 9600, 19200, 38400, 48000, 56000.

3.3.3 Gateway/ISDN Interface

The gateway/ISDN interface is provided through the TELEOS B100PC Communications Coprocessor NIC. The B100PC supports the following:

- Standard 8-pin ISDN RJ-45 interface
- Standard 6-pin analog phone RJ-11 interface
- ISDN BRI (2B+D); Line Rate, 192 Kbps (nominal)
- B channel data rate, 64Kbps
- D channel data rate, 16Kbps.

3.3.4 Man-Machine Interface

The MMI provides a menu-driven interface so the gateway operator can perform the following gateway functions:

- Gateway start-up
- Statistics gathering and reporting
- Gateway shutdown
- Address and routing table modifications.

The interface is provided through a standard Red, Blue, Green (RGB) color monitor and IBM-AT compatible keyboard.

3.4 SECURITY

No special security requirements have been identified for the multiprotocol gateway. Basic security protections defined for safe operation include password protection for gateway start up as well as shutdown. Modification of gateway addressing and routing information is performed only through off-line editing of appropriate data files. The gateway is designed in a structured manner to ensure easy adaptation of future security requirements such as cryptographic equipment.

4.0 GATEWAY SPECIFICATION

The functional operation of the multiprotocol gateway is shown in the eleven SDL diagrams contained in this section. The software modules depicted in each SDL diagrams will compose the gateway software.

4.1 GATEWAY DRIVER

The Gateway_Driver module is the "control" module for the gateway. It displays the gateway main menu, takes user inputs, and directs program control to the other modules, as appropriate. In addition to taking user inputs, it also accepts connection requests and data packets and causes the gateway to enter Call_Establishment and Transfer_Data modules, respectively.

Exhibit 4-1 depicts the operation of Gateway_Driver. The first process is the menu setup, which displays the gateway main menu and then prompts the user for an input. The gateway then places itself in an interrupt wait state, during which it is waiting to receive either a keyboard input showing which menu option is desired or a packet input. Upon receiving either of these interrupts, the gateway moves out of the interrupt wait state and directs gateway operation to another module, depending on the type of interrupt it has received. If the interrupt was a user-initiated keyboard interrupt, the driver transfers operation to either the Protocol_Start, Link_Answer_Setup, Statistics_Display, or Link_Shutdown modules. If the interrupt is a packet arrival, the driver transfers operation to either of two packet handling modules depending on packet type. If the incoming packet is a connection request packet, control is transferred to the Call_Establishment module. If the incoming packet is a data packet, control is transferred to the Transfer_Data_Packet.

4.2 PROTOCOL START

The Protocol_Start module, shown in Exhibit 4-2, loads the networking software for each of the three protocols supported by the gateway. It is invoked by the gateway operator in the Gateway_Driver module. The protocol software is loaded in the following order: X.25 NETBIOS, TCP/IP drivers, and ISDN NETBIOS. This software is typically supplied by vendors in an MS-DOS '.EXE' format and can be loaded by executing the '.EXE' file. The software for each protocol must be loaded into a unique memory segment within the COMPAQ 386 platform. The memory segment can usually be specified as an additional parameter during software loading. After each protocol is loaded, successful loading of the software is verified. If the the software has not been loaded correctly, an error message is generated and returned. At the conclusion of Protocol_Start execution, program control is returned to the Gateway_Driver module.

EXHIBIT 4-1
Gateway Driver Module

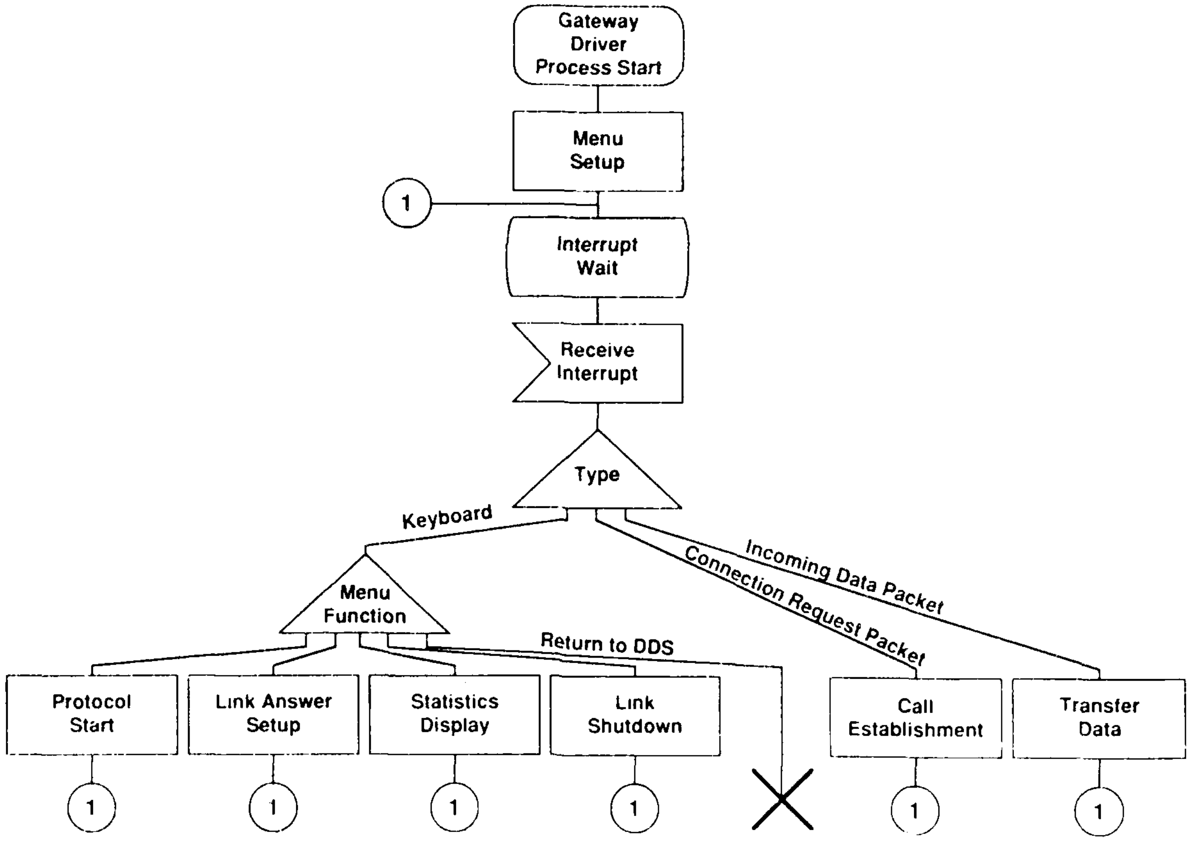
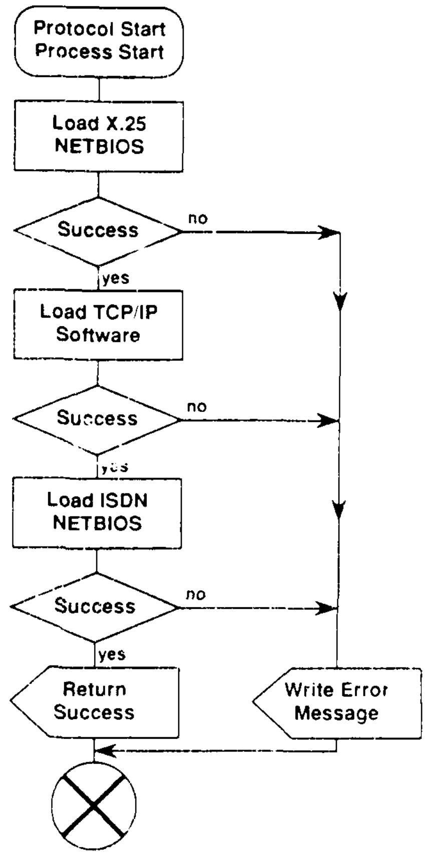


EXHIBIT 4-2
Protocol Start Module



4.3 LINK ANSWER SETUP

The Link_Answer_Setup provides a means of alerting the gateway of an Incoming Call_Request from one of the three ports (X.25, TCP/IP, or ISDN). The module posts interrupt routines in memory for each of the three boards. Exhibit 4-3 shows the operation of this routine. The module can be initially called as a menu option in the Gateway Driver; afterwards, the module is called whenever a connection hangup is detected.

The module starts by resetting all incoming call indication switches. There are switches for each of the three protocol networks, ISDN, X.25, and TCP/IP. Those switches alert the Driver routine of all Call Request appearances. Next, if the there is no interrupt routine outstanding for the X.25 board, the routine posts one. Similar decisions are made for the TCP/IP and ISDN boards. If any of the interrupt routines fail to be posted, the Link_Answer_Setup Module alerts the MMI.

4.4 CALL ESTABLISHMENT

The Call_Establishment module provides a mechanism for responding to requests for calls through the multiprotocol gateway. The routine drives the modules that post interrupt routines to accept all incoming Call_Requests, perform address extraction, look up the address translations in the Gateway Translation Matrix (GTM), and establish the data transmission interrupt routines. Exhibit 4-4 shows the flow of the Call_ Establishment Module. The modules that the module calls are as follows:

4.4.1 Extract_Address Module
4.4.2 Address_Lookup Module
4.4.3 Establish_Link Module
4.4.4 Data_Interrupt_Setup Module

Upon receiving a request for a call setup either from an X.25 network end user, a DDN end user or the ISDN link, the gateway determines the source and destination networks. Knowing the source network, the gateway can extract the destination node address from the packet. Once that address is translated to a legal address using the GTM, the gateway attempts to establish the link to the destination node. The source node is alerted if the call is accepted or rejected. If the call is successfully established, the gateway creates an interrupt routine to monitor both sides of the connection for incoming data packets.

4.4.1 Extract Address

The Extract_Address module, the first routine called by the Call_Answer module, accepts the incoming packet header and extracts the source and destination addresses. Exhibit 4-5 shows the processing required for both X.25 and TCP/IP packets. The module first extracts the length from the packet header, then

EXHIBIT 4-3
Link Answer Setup Module

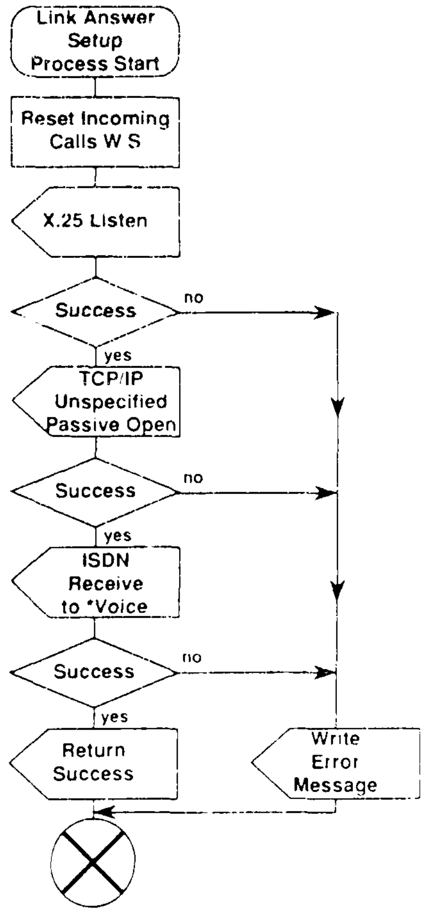


EXHIBIT 4-4
Call Establishment Module

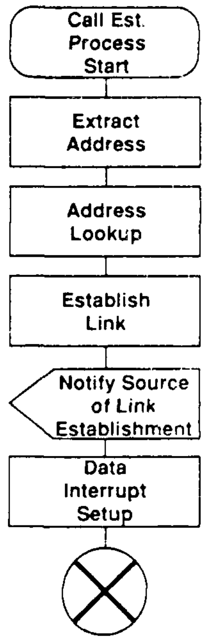
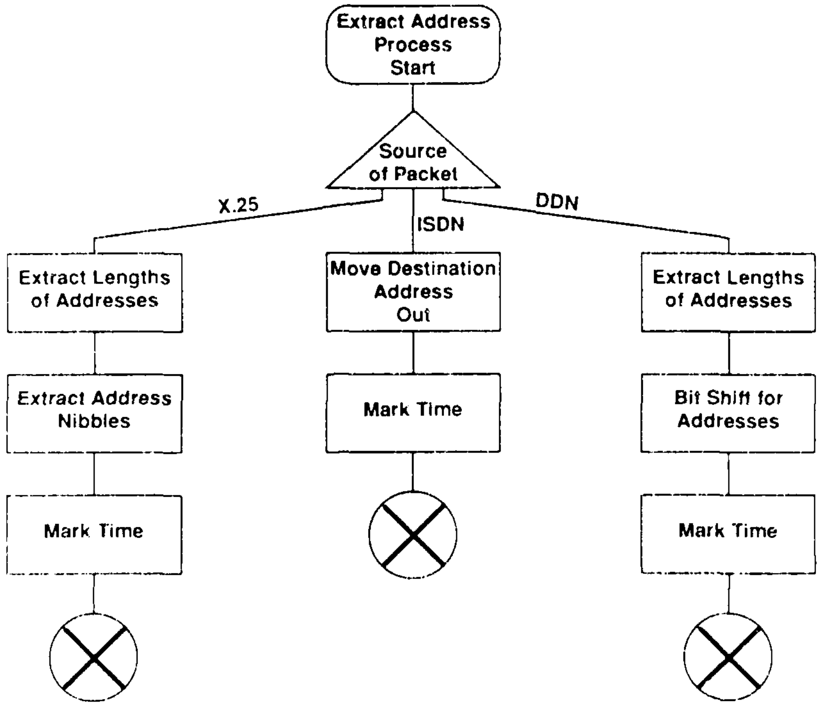


EXHIBIT 4-5
Extract Address Module



extracts the embedded source and destination addresses. With an ISDN packet, the address is passed through D channel call signaling. As a last step, the module initializes a connection clock that will be active for the duration of the call.

4.4.2 Address Lookup

The Address_Lookup module, called by the Call_Answer module, performs the actual GTM lookup for the translated destination address. The packet's embedded destination address will correctly route the packet to the destination node. Exhibit 4-6 shows the processing required for the Address_Lookup module.

To pass through a foreign network, the packet's original destination address must be altered to represent the conventions of the connecting network. If, for example, the destination address requires a connection through the ISDN, the gateway must establish a connection to the gateway between the ISDN and the destination node's network. The address translation could either require altering the destination address to an ISDN call number or, in the case of the gateway actually being a 5ESS switch hung off the ISDN, the gateway must prefix a 5ESS address on the destination address. The module scans the appropriate matrix column for the incoming destination address. Once it finds the correct matrix row, the module determines the coresponding destination address.

4.4.3 Establish Link

The Establish_Link module, after being called by the Call_ Answer module, attempts to complete the trans-gateway connections to the prescribed destination node. Exhibit 4-7 depicts the module's processing. Upon receiving the status of the connection, the module returns a success or failure message to the MMI. In the case of a connection through the ISDN, the module must bring up the various voice and data interfaces to the Teleos terminal adapter.

4.4.4 Data Interrupt Setup

The Data_Interrupt_Setup module, as shown in Exhibit 4-8, posts interrupt routines to notify the Gateway_Driver upon receipt of a data packet from either end of the established connection. When the Call_Answer module calls Data_Interrupt_ Setup, the routine attempts to set up listen requests to the network boards of the connection's two end points. The request is made either by a low level NETBIOS listen or by a higher level primitive. Finally the routine will return success if both sides complete the request without error; otherwise, the routine alerts the MMI as to the error type.

EXHIBIT 4-6
Address_Lookup

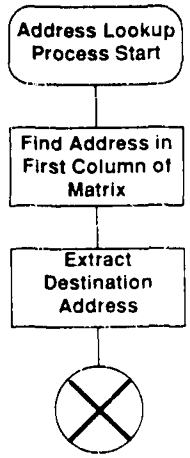


EXHIBIT 4-7
Establish Link Module

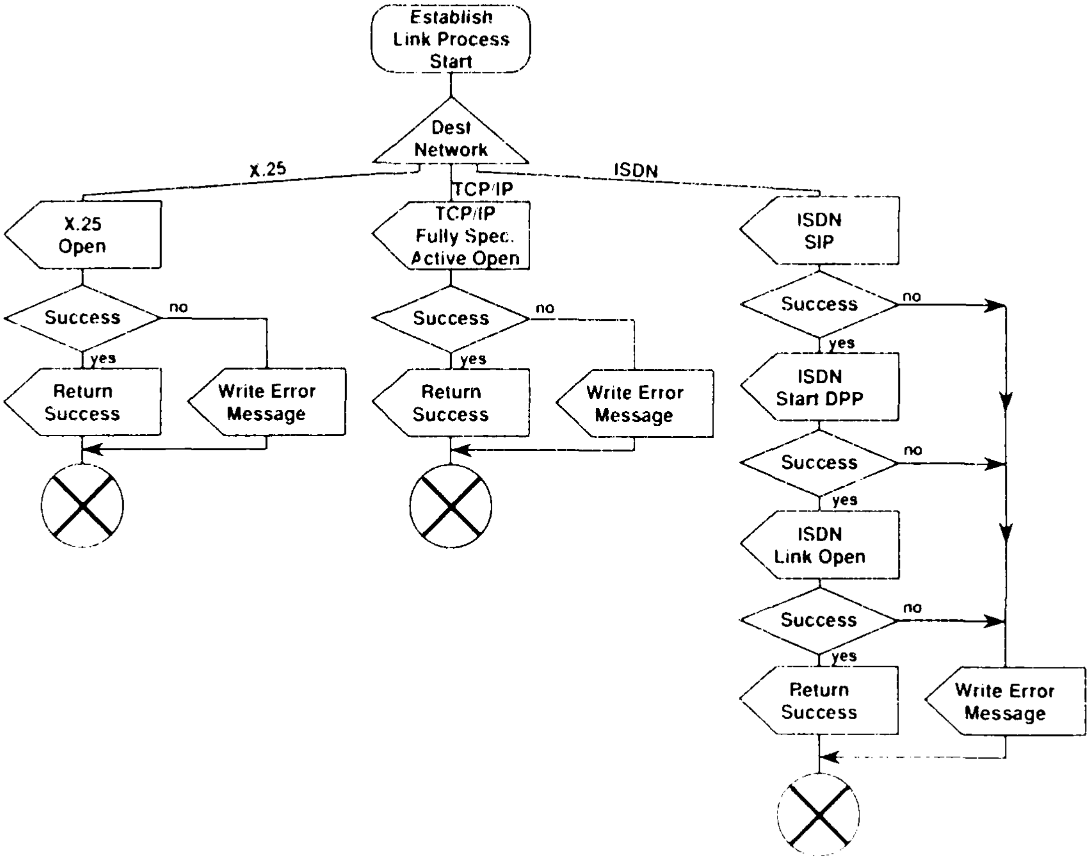
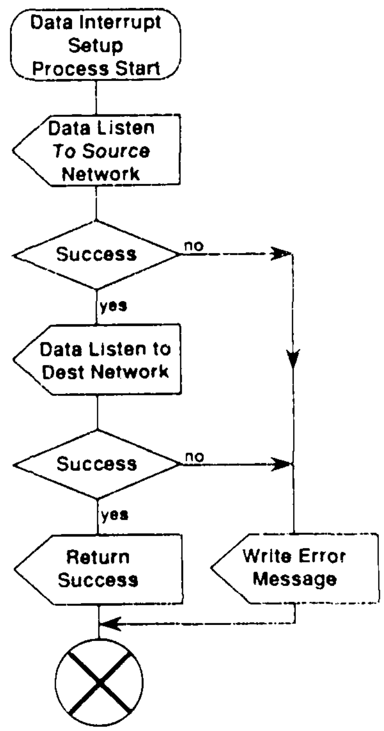


EXHIBIT 4-8
Data Interrupt Setup Module



4.5 TRANSFER DATA

The Transfer_Data module passes the packet traffic between the gateway 's end user nodes and accepts the close requests. The routine determines the source connection, performs any buffer translations or transfers, sends the packet data, updates statistics, and resets the call indication switch. Exhibit 4-9 shows the operations associated with the Transfer_Data module.

When the data packet is passed to the gateway, the interrupt service routine is alerted by the source network NETBIOS. The routine, in turn, turns on a global switch within the gateway process. The Transfer_Data module determines the source and type of the packet. If the source end user is sending a close notification, the gateway sends a hangup to the destination node and marks the connection as closed. If the source node is sending a data packet, the gateway extracts the data into a buffer and passes the buffer to the destination node. Once the packet has been transferred, the routine clears the call indication switch and updates the transmission statistics.

If the data packet originated from the ISDN end user (i.e., an X.25 or TCP/IP end user through the ISDN), the gateway must, in addition to checking for a data or a hangup packet type, check for a message packet type. The message packet is treated slightly different from the data packet; however, the net effect is the same. Because the structure requires that the data portion be an ASCII null-terminated string, the message packet must be disassembled into an output buffer before it is sent to the destination node.

4.6 STATISTICS DISPLAY

The Statistics_Display module displays, on the screen, the cumulative statistics that have been generated during the session in progress. It should be noted that the transmission statistics are not collected within this module; rather this module simply displays those statistics generated and updated in the Transfer_Data module.

The Statistics_Display operation is depicted in Exhibit 4-10. As shown, on invocation of Statistics_Display, a determination is made as to whether an X.25 or TCP/IP to ISDN session is in progress. Based on the type of session, the corresponding transmission statistics are displayed. If no session is currently active, transmission statistics will be displayed from the last active session. Once the statistics have been displayed, the Stats_Ind variable is updated to reflect this. The Statistics_Display module then enters a wait state in which it is waiting for operator indication that he/she has viewed the statistics and would like to return to the Gateway_Driver module.

EXHIBIT 4-9
Transfer Data Module

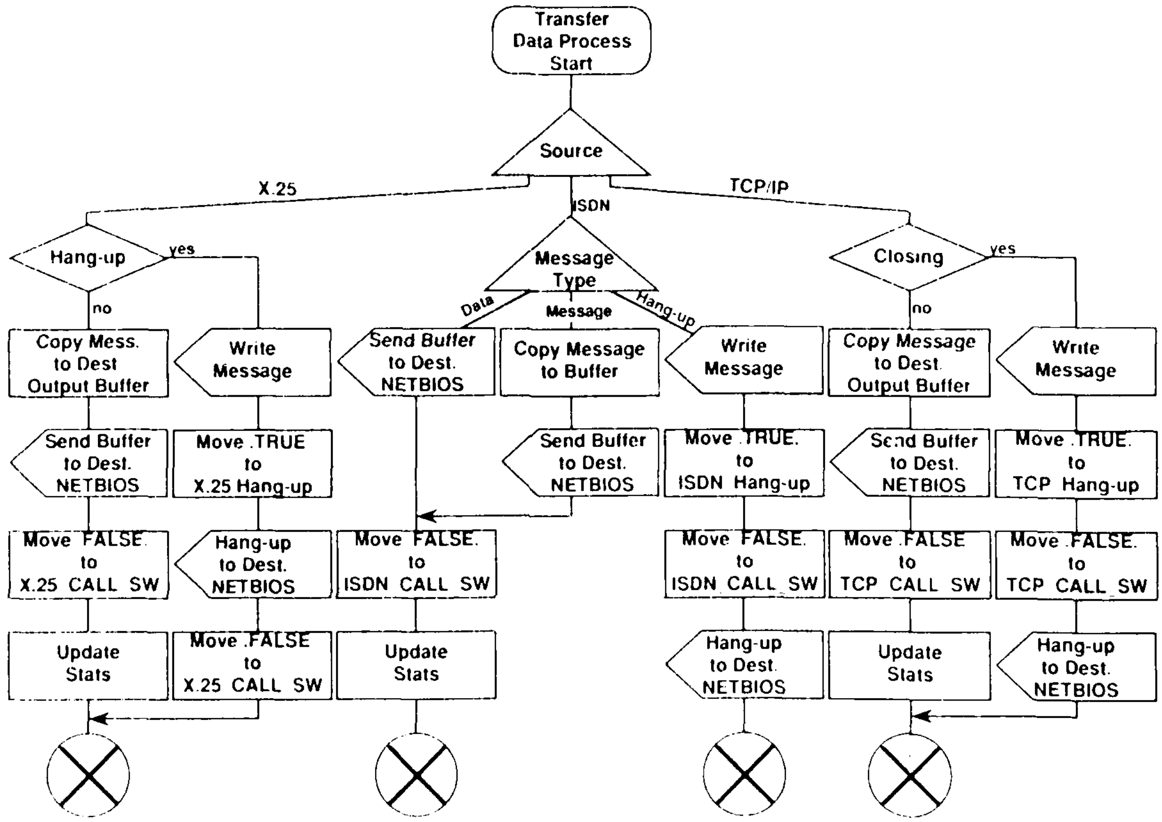
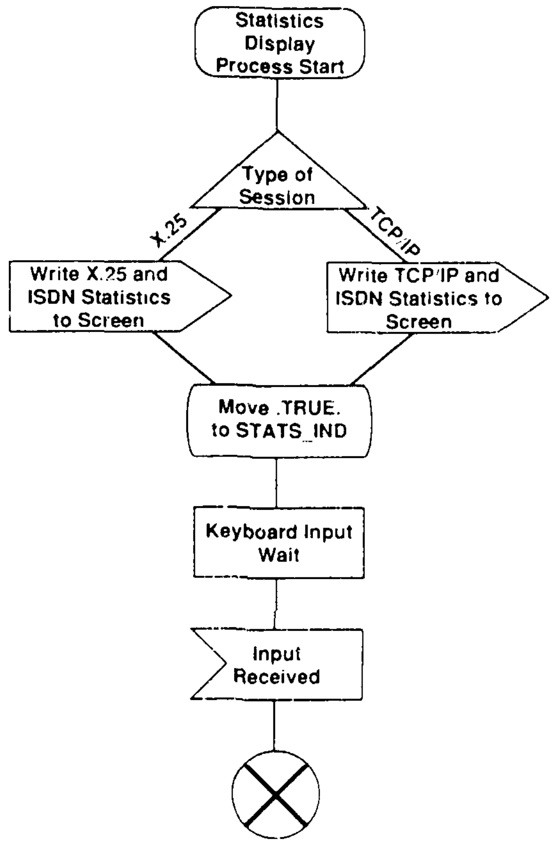


EXHIBIT 4-10
Statistics Display Module

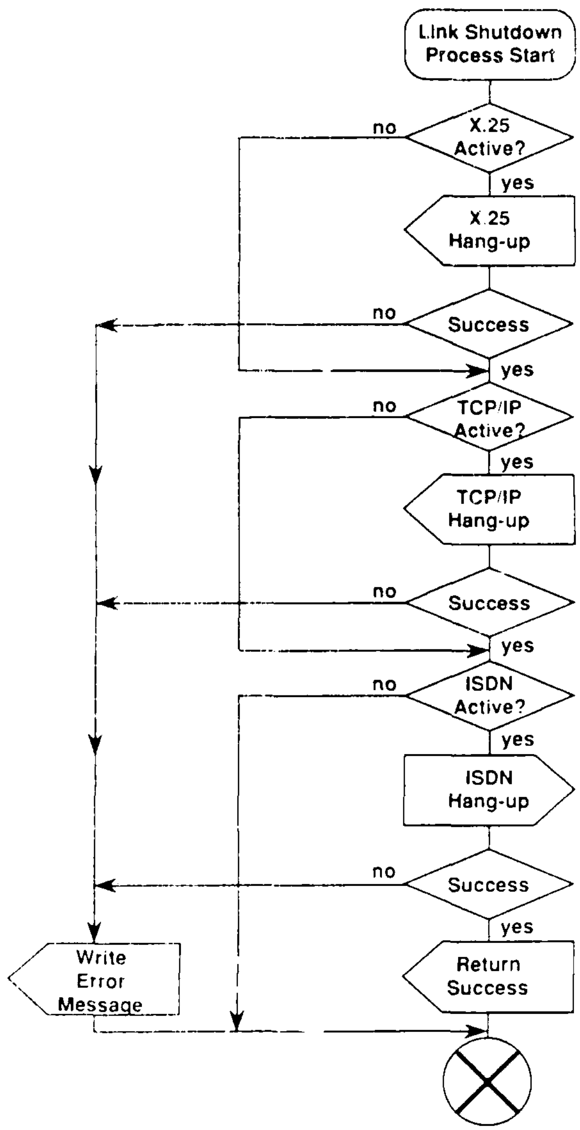


4.7 Link Shutdown

The Link_Shutdown module allows the gateway operator to gracefully terminate the current gateway session. This module is primarily used during gateway development, but may also prove useful in an operational environment if the gateway needs to be transported as it would allow current sessions to be disconnected. The module operation is shown in Exhibit 4-11. The procedure begins by determining whether an X.25 session is active. If so, an X.25 hangup is issued to terminate the session. The success of the X.25 hangup is then verified; if the hangup was not successful, an error message is generated and program control is returned to the Gateway Driver.

A similar procedure is followed for terminating the TCP/IP and ISDN session(s). The procedure begins by determining whether a TCP/IP or ISDN session is active. If so, a TCP/IP Closing or ISDN Hang-Up is issued to terminate the session. The success of the TCP/IP Closing or ISDN Hang-Up is then verified; if the Closing or Hang-Up was not successful, an error message is generated and program control is returned to the Gateway_ Driver.

EXHIBIT 4-11
Link Shutdown Module



A P P E N D I X   A

LIST OF ACRONYMS

ACRONYM	DESCRIPTION
BRI	Basic Rate Interface
bps	Bits per second
CCITT	The International Telegraph and Telephone Consultative Committee
DCE	Data circuit-terminating equipment
DTE	Data terminal equipment
DDN	Defense Data Network
DOD	Department of Defense
GTM	Gateway Translation Matrix
ISDN	Integrated Services Digital Network
IP	Internet Protocol
Kbps	Kilobits per second
KBps	Kilobytes per second
MPU	Master Processing Unit
MMI	Man-Machine Interface
NS/EP	National Security and Emergency Preparedness
NIC	Network Interface Card
OMNCS	Office Of The Manager, National Communications System
PC	Personal Computer
RAM	Random Access Memory
SDL	Specification and Description Language
TA	Terminal Adaptor
TCP	Transmission Control Protocol